

(19)



(11)

EP 4 273 670 B1

(12)

EUROPEAN PATENT SPECIFICATION

(45) Date of publication and mention of the grant of the patent:

04.06.2025 Bulletin 2025/23

(51) International Patent Classification (IPC):

G06F 3/01 (2006.01) **G06F 9/451** (2018.01)
G06F 3/0481 (2022.01) **G06F 3/048** (2013.01)
G06F 8/65 (2018.01)

(21) Application number: **22799838.2**

(52) Cooperative Patent Classification (CPC):

G06F 3/011; G06F 8/65; G06F 9/451

(22) Date of filing: **30.09.2022**

(86) International application number:

PCT/CN2022/123489

(87) International publication number:

WO 2023/173726 (21.09.2023 Gazette 2023/38)

(54) **INTERACTION METHOD AND APPARATUS, AND STORAGE MEDIUM**

INTERAKTIONSVERFAHREN UND -VORRICHTUNG SOWIE SPEICHERMEDIUM

PROCÉDÉ ET APPAREIL D'INTERACTION, ET SUPPORT DE STOCKAGE

(84) Designated Contracting States:

**AL AT BE BG CH CY CZ DE DK EE ES FI FR GB
GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO
PL PT RO RS SE SI SK SM TR**

(72) Inventor: **ZHOU, Chao**

Beijing 100086 (CN)

(30) Priority: **16.03.2022 CN 202210260731**

(74) Representative: **Dentons UK and Middle East LLP**

**One Fleet Place
London EC4M 7WS (GB)**

(43) Date of publication of application:

08.11.2023 Bulletin 2023/45

(56) References cited:

**CN-A- 109 782 920 CN-A- 111 586 452
CN-A- 111 656 256 CN-A- 114 625 253
US-A1- 2018 359 448 US-A1- 2020 043 243**

(73) Proprietor: **Beijing Zitiao Network Technology Co., Ltd.
Beijing 100190 (CN)**

Note: Within nine months of the publication of the mention of the grant of the European patent in the European Patent Bulletin, any person may give notice to the European Patent Office of opposition to that patent, in accordance with the Implementing Regulations. Notice of opposition shall not be deemed to have been filed until the opposition fee has been paid. (Art. 99(1) European Patent Convention).

Description

TECHNICAL FIELD

[0001] The present disclosure relates to virtual reality technology, and in particular, to an interaction method and apparatus, and a storage medium.

BACKGROUND

[0002] Virtual reality (VR) technology is a virtual environment created by modern computer technology. Users can use a specific human-computer interaction device and apparatus to interact with the virtual environment to generate an immersive experience. Extended reality (XR) is a further development of virtual reality technology. The extended reality refers to the creation of a real and virtual combined and human-computer interactive environment through computer technology and a wearable device.

[0003] Interaction with a virtual object is an essential part of a XR world. There are two interaction methods according to distance: one is near-field interaction, where the user touches an object through a finger click or with a handle or the like; and the other is far-field interaction, where the user generally uses ray to perform collision detection on the virtual object to further complete an entire trigger cycle.

[0004] For the far-field interaction, more than one input device may exist at the same time in an XR scenario, and a user interface (UI) in the scenario can only respond to one trigger at the same time. Therefore, how to manage interaction between multiple input devices and the UI in the XR scenario has become an urgent problem to be solved.

[0005] US20180359448A1 discloses systems and methods for true interaction between a plurality of users (e.g., and their respective avatars) in real-time within a virtual reality environment. The disclosed system can provide virtual, physicalized objects (e.g., virtual object representations) that can be handed around among users (avatars), jointly lifted, carried and placed, while maintaining a smooth continuity of experience and plausible physical simulation that comports with a person's typical and expected experience of a given physical environment. Furthermore, it can achieve this even with low and variable bandwidth between users, or users and a central communication server.

SUMMARY

[0006] The present disclosure provides an interaction method and apparatus, and a storage medium, to manage the interaction between multiple input devices and a UI in an XR scenario. The present disclosure is defined in the independent claims and further embodiments are defined in the dependent claims.

BRIEF DESCRIPTION OF DRAWINGS

[0007] In order to more clearly illustrate the embodiments of the present disclosure or the technical solutions in the prior art, the following briefly introduces drawings needed to be used in the description of the embodiments or the prior art. Obviously, the drawings in the following description are only some embodiments of the present disclosure, and for those of ordinary skill in the art, other drawings can also be obtained according to these drawings without any creative effort.

FIG. 1 is a schematic diagram of an interaction system architecture provided by an embodiment of the present disclosure.

FIG. 2 is a schematic flowchart of an interaction method provided by an embodiment of the present disclosure.

FIG. 3 is a schematic diagram of a hover event processing provided by an embodiment of the present disclosure.

FIG. 4 is a schematic diagram of a trigger event processing provided by an embodiment of the present disclosure.

FIG. 5 is a schematic flowchart of another interaction method provided by an embodiment of the present disclosure.

FIG. 6 is a schematic flowchart of yet another interaction method provided by an embodiment of the present disclosure.

FIG. 7 is a schematic structural diagram of an interaction apparatus provided by an embodiment of the present disclosure.

FIG. 8 is a schematic diagram of a basic hardware architecture of a target execution component provided by an embodiment of the present disclosure.

DESCRIPTION OF EMBODIMENTS

[0008] The technical solutions in the embodiments of the present disclosure will be clearly and completely described below with reference to the drawings in the embodiments of the present disclosure. Obviously, the described embodiments are only a part of the embodiments of the present disclosure, but not all of the embodiments. Based on the embodiments in the present disclosure, all other embodiments obtained by those of ordinary skill in the art without creative effort shall fall within the protection scope of the present disclosure.

[0009] The terms "first", "second", "third", "fourth", etc. (if present) in the description and claims of the present disclosure as well as the drawings are used to distinguish similar objects and are not necessarily used to describe a specific order or sequence. It is to be understood that the data so used may be interchanged under appropriate circumstances such that the embodiments of the present disclosure described herein can be practiced in sequences other than those illustrated or described herein.

Furthermore, the terms "including" and "comprising" and any variations thereof are intended to cover non-exclusive inclusion. For example, a process, a method, a system, a product or a device including a series of steps or units is not necessarily limited to those expressly listed steps or units, but also may include other steps or units not expressly listed or inherent to the process, the method, the product or the device.

[0010] XR technology includes augmented reality (AR), VR, and mixed reality (MR). It integrates virtual content and a real scenario through a hardware device and multiple technical means. In an XR scenario, there are two interaction methods according to distance: one is near-field interaction, where the user touches an object through a finger click or with a process; and the other is far-field interaction, where the user generally uses ray to perform collision detection on the virtual object to further complete an entire trigger cycle. Where, for the latter, more than one input device may exist at the same time in the XR scenario, and a UI in the scenario can only respond to one trigger at the same time. Therefore, a mechanism is needed to manage interaction between multiple input devices and the UI.

[0011] Therefore, an embodiment of the present disclosure proposes an interaction method, in which a trigger request sent by an input device is received through an execution component, and the execution component corresponds to the input device. The state of the execution component is active and the states of other execution components are inactive, which makes only one input device be active (can trigger events) at the same time, so that the UI only responds to one trigger at the same time. And then the trigger signal collision detection and event processing are performed according to the trigger request, and the processed result is displayed on the UI to ensure correct, fast and complete interaction between multiple input devices and the UI in the XR scenario.

[0012] Optionally, the interaction method provided by the embodiment of the present disclosure may be applied to an interaction system shown in FIG. 1. In FIG. 1, the interaction system architecture may include an execution component 101. The execution component 101 is one of a plurality of execution components, and one execution component corresponds to one input device. Here, it can be understood that an execution component is bound to an input device, and the execution component only processes the relevant information sent by the input device bound to it. For example, there are execution components 1, 2, and 3 and input devices A, B, and C, and the execution component 1 corresponds to the input device A. The execution component 1 receives a trigger request sent by the input device A, and can perform trigger signal collision detection and event processing and so on based on the trigger request.

[0013] In a specific implementation, the execution component 101 receives a trigger request sent by the corresponding input device, performs trigger signal collision detection and event processing based on the trigger

request, and displays the processed result on the UI, which solves the problem of how the UI responds correctly for interaction between multiple input devices and the UI in the XR scenario. The state of the execution component 101 is active, and the states of other execution components are inactive, so that only one input device is active (can trigger events) at the same time, and the UI only responds to one trigger at the same time.

[0014] In addition, the interaction system architecture may include a management system 102, and the execution component 101 may interact with the management system 102 to manage interaction between the multiple input devices and the UI.

[0015] The management system 102 may receive each trigger request of each input device. If the user sends a trigger request through an input device, the management system 102 receives the trigger request, and manages multiple execution components according to the trigger request. For example, the management system 102 sets the state of the execution component corresponding to the input device to be active, and sets the states of other execution components to be inactive, and so on. This makes only one input device is active at the same time, so that the UI only responds to one trigger at the same time. Then, the management system 102 may also sends the trigger request to the execution component corresponding to the input device. For example, the execution component is the execution component 101, and the execution component 101 performs trigger signal collision detection and event processing based on the trigger request. This ensures correct, fast and complete interaction between multiple input devices and the UI in the XR scenario.

[0016] It should be noted that the system architectures and the business scenarios described in the embodiments of the present disclosure are for the purpose of illustrating the technical solutions of the embodiments of the present disclosure more clearly, and do not constitute a limitation on the technical solutions provided by the embodiments of the present disclosure. Those of ordinary skill in the art know that, with the evolution of the system architectures and the emergence of new business scenarios, the technical solutions provided by the embodiments of the present disclosure are also applicable to similar technical problems.

[0017] The technical solutions of the present disclosure are described below by taking several embodiments as examples, and the same or similar concepts or processes may not be repeated in some embodiments.

[0018] FIG. 2 is a schematic flowchart of an interaction method provided by an embodiment of the present disclosure. The execution body of this embodiment takes the execution component in FIG. 1 as an example, and the specific execution body can be determined according to the actual application scenario, and the embodiment of the present disclosure does not specifically limit to this. For the convenience of description, the execution component in FIG. 2 can be understood as a target execution

component, and the target execution component is one of a plurality of execution components, and one execution component corresponds to one input device. As shown in FIG. 2, the interaction method provided by the embodiment of the present disclosure may include the following.

[0019] S201: receive a trigger request, where the trigger request is sent by a target input device, the target execution component corresponds to the target input device, a state of the target execution component is active, and a state of each remaining execution component in the plurality of execution components except the target execution component is set to be inactive.

[0020] Here, the user can send a trigger request through a target input device, the target input device can be any input device, the input device corresponds to an execution component, that is, is bound to an execution component, and the execution component processes the relevant information sent by the target input device.

[0021] The trigger request may be a trigger request received by the target execution component from the management system. That is, the management system receives the trigger request sent by the target input device, and then forwards the trigger request to the target execution component, so that the target execution component can accurately process the relevant information sent by the corresponding input device. Here, the management system can manage a plurality of execution components including the target execution component, for example, the controlling forwarding corresponding information to each execution component, or controlling the state of the execution component to set the state of the target execution component to be active and set the state of each remaining execution component except the target execution component to be inactive. So that only one input device is active (can trigger events) at the same time, and the UI only responds to one trigger at the same time.

[0022] Here, the trigger request may carry an identifier of the target input device. Taking two input devices as an example, if the two input devices are a left handle and a right handle, respectively, and the target input device is the right handle, the trigger request may carry the identifier of the right handle. The management system receives each trigger request from each input device. After receiving the trigger request from the target input device, the management system may determine the target execution component corresponding to the target input device according to the identifier of the target input device carried in the trigger request, and then set the state of the target execution component to be active, and set the state of each remaining execution component except the target execution component to be inactive. So that only one input device is active (can trigger events) at the same time, and the UI only responds to one trigger at the same time to meet the actual application needs.

[0023] In the embodiment of the present disclosure,

the management system may pre-store a corresponding relationship between the identifiers of the input devices and the execution components. In this way, after receiving the trigger request of the target input device, the target execution component corresponding to the target input device is determined according to the corresponding relationship and the trigger request carrying the identifier of the target input device, which is simple and convenient. The identifier may be a name and a serial number of the input device and other information that can uniquely identify the identity of the device.

[0024] In addition, the input device corresponding to the execution component in the inactive state is triggered. For example, the execution component corresponding to the left handle is in an inactive state, and the execution component corresponding to the right handle is in an active state. When the left handle is triggered, the management system will first cancel and clear all trigger events of the currently active handle (right handle), and, at the same time, set the execution component corresponding to the left handle to be active to respond to the corresponding trigger to meet the actual application needs. The user is unaware of whether the handle is active. This improves the user experience.

[0025] S202: cache the trigger request in a preset action queue, and perform trigger signal collision detection based on a UI after receiving an update instruction.

[0026] Here, the target execution component caches the trigger request in the preset action queue. On the one hand, it can manage the trigger requests with a mutually exclusive relationship to ensure that the two action cycles for the trigger requests are not overlapped. On the other hand, within the action cycle, even if the trigger request is not sent every frame, it can ensure that the execution component generates a trigger event every frame, so that the UI can completely process the corresponding event.

[0027] In the embodiment of the present disclosure, the trigger signal may be a collision trigger signal, such as ray. After receiving the update instruction, the target execution component can track the position and the attitude of the target input device to determine the starting point and the direction of the ray, update the position and the direction of the ray, etc., and perform ray collision detection in each frame to determine the node where the ray collides with the UI, and display the node, where a collision is occurred, on the UI, so that the user can know the state of the input device.

[0028] In addition, after determining the node where the trigger signal collides with the UI, the target execution component may also store the node where the trigger signal collides with the UI, so as to facilitate subsequent processing of related events based on the stored node where a collision is occurred.

[0029] S203: obtain, according to a preset sequence, a trigger request from the preset action queue as a trigger request to be executed, after the trigger signal collision detection is performed.

[0030] Here, the preset sequence can be set according

to the actual situation. For example, it is first-in-first-out, that is, after the trigger signal collision is performed, the target execution component obtains a trigger request from the preset action queue in the sequence of first-in-first-out as a trigger request to be executed, so as to meet various application needs.

[0031] As an example, taking the trigger signal as ray as an example, as shown in FIG. 3, after ray collision detection is performed, and before obtaining, according to a preset sequence, a trigger request from a preset action queue as a trigger request to be executed, the target execution component can also update the length of ray rendering, update the display and hide and the position of an intersection cursor of the ray and the UI, and when the state of the target execution component is active, call the corresponding hover processing interface to process the hover event according to the intersection cursor of the ray and the UI in the previous frame detected through the trigger signal collision detection and according to the intersection cursor of the ray and the UI in the current frame detected through the trigger signal collision detection, and display the processed result of the hover processing interface on the UI. This solves the hover event processing for interaction between multiple input devices and the UI in the XR scenario.

[0032] The hover processing interface includes a hover entry interface, a hovering stop interface, and a hovering end interface. The target execution component can compare the node where the trigger signal collides with the UI in the previous frame with the node where the trigger signal collides with the UI in the current frame. According to a node, in a the comparison result, which is not among the nodes where the trigger signal collides with the UI in the previous frame, but is among the nodes where the trigger signal collides with the UI in the current frame, the hover entry interface is called to process the hover event; according to a node, in a the comparison result, which is among the nodes where the trigger signal collides with the UI in the previous frame and is among the nodes where the trigger signal collides with the UI in the current frame, the hover stop interface is called to process the hover event; and according to a node, in a the comparison result, which is among the nodes where the trigger signal collides with the UI in the previous frame, but is not among the nodes where the trigger signal collides with the UI in the current frame, the hover end interface is called to process the hover event. That is, different events call different interface processing, so as to successfully process the corresponding event through the called interface. For example, the nodes where the trigger signal collides with the UI in the previous frame includes A, B, and C, and the nodes where the trigger signal collides with the UI in the current frame includes A, C, and D, then the hover interface is called on A and C to process the hover event, the hover end interface is called on B to process the hover event, and the hover entry interface is called on D to process the hover event.

[0033] S204: call, according to the trigger request to be

executed, a corresponding action execution interface to process a trigger event, and display a processed result of the action execution interface on the UI.

[0034] As an example, as shown in FIG. 4, before calling, according to the trigger request to be executed, the corresponding action execution interface to process the trigger event, the target execution component may also determine whether the action triggered by the trigger request currently executed is within the action cycle. If the action triggered by the trigger request currently executed is not within the action cycle, it means that the action triggered by the trigger request currently executed may be executed or the user has sent an end request, and it is no mutual exclusion with the action triggered by the trigger request to be executed. At this time, the target execution component can call the corresponding action execution interface to process the trigger event according to the trigger request to be executed.

[0035] Here, if the action triggered by the trigger request currently executed is within the action cycle, the target execution component can further determine whether the action triggered by the trigger request to be executed and the action triggered by the trigger request currently executed are mutually exclusive. If it is not mutually exclusive, the target execution component calls the corresponding action execution interface to process the trigger event according to the trigger request to be executed. If it is mutually exclusive, after the execution of the action triggered by the trigger request currently executed is completed, the target execution component calls the corresponding action execution interface to process the trigger event according to the trigger request to be executed.

[0036] For non-mutually exclusive trigger requests, the target execution components can perform processing in parallel to meet application needs. For mutually exclusive trigger requests, the target execution component processes another trigger request after processing the first trigger request to ensure the integrity of the action cycle.

[0037] In the embodiment of the present disclosure, when determining whether the action triggered by the trigger request to be executed and the action triggered by the trigger request currently executed are mutually exclusive, the target execution component can obtain the type of the action triggered by the trigger request to be executed and the type of the action triggered by the trigger request currently executed, and then determine whether the type of the action triggered by the trigger request to be executed is consistent with the type of the action triggered by the trigger request currently executed. If they are consistent, it is determined that the action triggered by the trigger request to be executed and the action triggered by the trigger request currently executed are not mutually exclusive. If they are inconsistent, it is determined that the action triggered by the trigger request to be executed and the action triggered by the trigger request currently executed are mutually exclusive. The subsequent steps are executed based on the

determining result to ensure that subsequent processing proceeds normally. The trigger request may carry the type of the triggered action, and the target execution component obtains the type of the action triggered in the trigger request to be executed and the type of the action triggered in the trigger request currently executed, according to the type of the action carried in the trigger request to be executed and the type of the action carried in the trigger request currently executed.

[0038] In addition, the trigger request may carry an activation identifier, where the activation identifier is used to indicate whether the action triggered by the trigger request is an activation action. When determining whether the action triggered by the trigger request currently executed is within the action cycle, the target execution component can determine whether the action triggered by the trigger request currently executed is an activation action according to the activation identifier carried in the trigger request currently executed. If the action triggered by the trigger request currently executed is an activation action, the target execution component determines that the action triggered by the trigger request currently executed is within the action cycle. For example, taking the button on the handle as an example, the user sends a trigger request by pressing the button, the trigger request carries an activation identifier, and the activation identifier is used to indicate that the action triggered by the trigger request is an activation action (that is, when the button is pressed, the activation identifier carried in the trigger request indicates that the action triggered by the trigger request is an activation action; while the button is lifted, the activation identifier carried in the trigger request indicates that the action triggered by the trigger request is not an activation action). The management system receives the trigger request, and sends the trigger request to the corresponding execution component, and the execution component determines whether the action triggered by the trigger request is an activation action according to the activation identifier, so as to accurately determine whether the action triggered by the trigger request is with the action cycle.

[0039] In the embodiment of the present disclosure, the target execution component caches received trigger requests in a queue, and then takes out a trigger request from the queue as a trigger request to be executed for processing. Under the condition of ensuring the integrity of the action cycle, the target execution component calls the corresponding action execution interface to process the trigger event, and displays the processed result of the action execution interface on the UI.

[0040] When calling the corresponding action execution interface to process the trigger event, the target execution component can obtain the type of the action triggered by the trigger request to be executed. And then, according to the type, the target execution component determines the corresponding action execution interface to be called, and then calls the corresponding action execution interface to process the trigger event to ensure

accurate subsequent processing. Here, the types may include click, joystick, and the like. The target execution component determines the action execution interface as click, joystick or the like according to the type, so as to call the action execution interface of click, joystick or the like to process the event such as the click, joystick or the like.

[0041] Here, the action execution interface may include an interactive component such as a UI component with specific function, like a common button (Button), a progress bar (Slider), a single check button (Tab), a scroll view (Scroll View), and the like, so as to implement the respective functional logic in the interface, respond to the trigger of the input device, and open up the complete UI interaction link.

[0042] In the embodiment of the present disclosure, a trigger request sent by an input device is received through an execution component, where the execution component corresponds to the input device. A state of the execution component is active, and states of other execution components are inactive. Thus, only one input device is active (can trigger an event) at the same time, and a UI only responds to one trigger at the same time. And then, according to the trigger request, trigger signal collision detection and event processing are performed, and a processed result is displayed on the UI. It solves the problem that how the UI responds correctly for the interaction between multiple input devices and the UI in a XR scenario, and ensures the UI interaction in the XR scenario is correct, fast and complete. Moreover, in the embodiment of the present disclosure, after receiving the trigger request sent by the management system, the execution component caches the trigger request in a preset action queue. In this way, on the one hand, it can manage the trigger requests with a mutually exclusive relationship to ensure that the two action cycles for the trigger requests are not overlapped. On the other hand, within the action cycle, even if the trigger request is not sent every frame, it can ensure that the execution component generates a trigger event every frame, so that the UI can completely process the corresponding event.

[0043] In addition, taking an interaction between a target execution component and a management system, the management system receiving a trigger request sent by a target input device and forwarding the trigger request to the target execution component as an example, FIG. 5 provides a schematic flowchart of another interaction method, that is, the flow of an interaction method corresponding to the management system. Where the relevant description refers to FIG. 2, which is not repeated here. As shown in FIG. 5, the method may include the following.

[0044] S501: receive a trigger request from a target input device.

[0045] S502: according to the trigger request, set the state of the target execution component to be active, and set the state of each of the remaining execution components of a plurality of managed execution components except the target execution component to be inactive,

where an execution component corresponds to an input device, and the target execution component corresponds to the target input device.

[0046] Here, one input device corresponds to one execution component, that is, one input device is bound to one execution component. In this way, the management system can set the state of the target execution component corresponding to the target input device to be active, and set the states of other execution components to be inactive. So that only one input device is active (can trigger events) at the same time, and the UI only responds to one trigger at the same time to meet the actual application needs.

[0047] S503: send the trigger request to the target execution component, where the trigger request is used to instruct the target execution component to cache the trigger request in a preset action queue, and perform the trigger signal collision detection based on the UI after receiving an update instruction; and after the trigger signal collision detection is performed, according to a preset sequence, obtain a trigger request from the preset action queue as the trigger request to be executed, call the corresponding action execution interface to process the trigger event according to the trigger request to be executed, and display the processed result of the action execution interface on the UI.

[0048] After setting the state of the execution component to be active, the management system sends the trigger request to the target execution component, so that the target execution component caches the received trigger request in a queue, then takes out a trigger request from the queue as a trigger request to be executed for processing, under the condition of ensuring the integrity of the action cycle, calls a corresponding action execution interface to process the trigger event, and displays the processed result of the action execution interface on the UI.

[0049] In the embodiment of the present disclosure, through interacting with multiple execution components, the management system manages the interaction between multiple input devices and the UI. Where one execution component corresponds to one input device. After receiving a trigger request from an input device, the management system sets the state of the execution component corresponding to the input device to be active, and sets the states of other execution components to be inactive. So that only one input device is active (can trigger events) at the same time. And then the management system sends the trigger request to the corresponding execution component. The execution component performs trigger signal collision detection and event processing, and displays the processed results on the UI to ensure interaction between multiple input devices and the UI in the XR scenario are correct, quick and complete.

[0050] In addition, FIG. 6 is a schematic flowchart of yet another interaction method provided by an embodiment of the present disclosure. This embodiment describes the process of the interaction method from the interaction

between the management system and the execution component. As shown in FIG. 6, the method may include the following.

[0051] S601: the management system receives a trigger request from a target input device.

[0052] S602: the management system sets the state of the target execution component to be active according to the trigger request, and sets the state of each of the remaining execution components of a plurality of managed execution components except the target execution component to be inactive, where an execution component corresponds to one input device, and the target execution component corresponds to the target input device.

[0053] S603: send the trigger request to the target execution component.

[0054] S604: the target execution component receives the trigger request, caches the trigger request in a preset action queue, and performs trigger signal collision detection based on the UI after receiving an update instruction.

[0055] The target execution component caches the trigger request in the preset action queue. On the one hand, it can manage the trigger requests with a mutually exclusive relationship to ensure that the two action cycles for the trigger requests are not overlapped. On the other hand, within an action cycle, even if the trigger request is not sent every frame, it can ensure that the execution component generates a trigger event every frame.

[0056] S605: after the trigger signal collision detection is performed, the target execution component obtains, according to a preset sequence, a trigger request from the preset action queue as a trigger request to be executed.

[0057] Here, after the trigger signal collision detection is performed, and before obtaining, according to the preset sequence, a trigger request from the preset action queue as the trigger request to be executed, the target execution component may also call a corresponding hover processing interface to process a hover event according to a node where a trigger signal collides with the UI in a previous frame and which is detected through the trigger signal collision detection and according to a node where the trigger signal collides with the UI in a current frame and which is detected through the trigger signal collision detection, and display a processed result of the hover processing interface on the UI.

[0058] The hover processing interface includes a hover entry interface, a hover stop interface, and a hover end interface. The target execution component can compare the node where the trigger signal collides with the UI in the previous frame and the node where the trigger signal collides with the UI in the current frame. According to a node, in a the comparison result, which is not among the nodes where the trigger signal collides with the UI in the previous frame, but is among the nodes where the trigger signal collides with the UI in the current frame, the hover entry interface is called to process the hover event; according to a node, in a the comparison result, which is

among the nodes where the trigger signal collides with the UI in the previous frame and is among the nodes where the trigger signal collides with the UI in the current frame, the hover stop interface is called to process the hover event; and according to a node, in a the comparison result, which is among the nodes where the trigger signal collides with the UI in the previous frame, but is not among the nodes where the trigger signal collides with the UI in the current frame, the hover end interface is called to process the hover event.

[0059] S606: the target execution component calls, according to the trigger request to be executed, a corresponding action execution interface to process a trigger event, and displays a processed result of the action execution interface on the UI.

[0060] The target execution component can first determines whether the action triggered by the trigger request currently executed is within an action cycle. If the action triggered by the trigger request currently executed is not within the action cycle, the target execution component can call, according to the trigger request to be executed, the corresponding action execution interface to process the trigger event.

[0061] Here, if the action triggered by the trigger request currently executed is within the action cycle, the target execution component can further determine whether the action triggered by the trigger request to be executed and the action triggered by the trigger request currently executed are mutually exclusive. If it is not mutually exclusive, the target execution component calls the corresponding action execution interface to process the trigger event according to the trigger request to be executed. If it is mutually exclusive, after the execution of the action triggered by the trigger request currently executed is completed, the target execution component calls, according to the trigger request to be executed, the corresponding action execution interface to process the trigger event.

[0062] In the embodiment of the present disclosure, the target execution component may obtain the type of the action triggered by the trigger request to be executed and the type of the action triggered by the trigger request currently executed, and determine whether the action triggered by the trigger request to be executed and the action triggered by the trigger request currently executed are mutually exclusive by determining whether the two types are consistent. If the type of the action triggered by the trigger request to be executed is consistent with the type of the action triggered by the trigger request currently executed, the target execution component determines that the action triggered by the trigger request to be executed and the action triggered by the trigger request currently executed are not mutually exclusive; otherwise, it is determined that the action triggered by the trigger request to be executed and the action triggered by the trigger request currently executed are mutually exclusive.

[0063] In addition, the trigger request carries an activa-

tion identifier, and the activation identifier is used to indicate whether the action triggered by the trigger request is an activation action.

[0064] When determining whether the action triggered by the trigger request currently executed is within the action cycle, the target execution component can determine whether the action triggered by the trigger request currently executed is an activation action according to the activation identifier carried in the trigger request currently executed. If the action triggered by the trigger request currently executed is an activation action, the target execution component determines that the action triggered by the trigger request currently executed is within the action cycle.

[0065] When calling the corresponding action execution interface to process the trigger event, the target execution component can obtain the type of the action triggered by the trigger request to be executed. And then, according to the type, the target execution component determines the corresponding action execution interface to be called, and then calls the corresponding action execution interface to process the trigger event.

[0066] To sum up, the embodiment of the present disclosure, through interacting with multiple execution components, the management system manages the interaction between multiple input devices and the UI. Where one execution component corresponds to one input device. After receiving a trigger request from an input device, the management system sets the state of the execution component corresponding to the input device to be active, and sets the states of other execution components to be inactive. So that only one input device is active (can trigger events) at the same time and the UI only responds to one trigger at the same time. Then, the management system sends the trigger request to the execution component corresponding to the input device. After receiving the trigger request sent by the management system, the execution component caches the trigger request in a preset action queue. Whereby, it can manage the trigger requests with a mutually exclusive relationship to ensure that the two action cycles for the trigger requests are not overlapped. On the other hand, within the action cycle, even if the trigger request is not sent every frame, it can ensure that the execution component generates a trigger event every frame. In addition, the execution component performs trigger signal collision detection and event processing, and displays the processing results on the UI to ensure that the UI interaction in the XR scenario is correct, fast and complete.

[0067] Corresponding to the interaction method in the embodiment, FIG. 7 is a schematic structural diagram of an interaction apparatus 70 provided by an embodiment of the present disclosure. For convenience of explanation, only the parts related to the embodiment of the present disclosure are shown. The interaction apparatus 70 includes: a receiving module 701, a processing module 702, an obtaining module 703, and a calling module 704. The interaction apparatus 70 here may be the target

execution component itself, or a chip or an integrated circuit that implements the function of the target execution component. It should be noted here that the division of the receiving module, the processing module, the obtaining module and the calling module is only a division of logical functions, two of which may be integrated or independent in physical.

[0068] The receiving module 701 is configured to receive a trigger request, where the trigger request is sent by a target input device, the target execution component corresponds to the target input device, a state of the target execution component is active, and a state of each remaining execution component in the plurality of execution components except the target execution component is inactive.

[0069] The processing module 702 is configured to cache the trigger request in a preset action queue, and perform trigger signal collision detection based on a UI after receiving the update instruction.

[0070] The obtaining module 703 is configured to obtain, according to a preset sequence, a trigger request from the preset action queue as a trigger request to be executed, after the trigger signal collision detection is performed.

[0071] The calling module 704 is configured to call, according to the trigger request to be executed, a corresponding action execution interface to process a trigger event, and display a processed result of the action execution interface on the UI.

[0072] In a possible implementation, the calling module 704 is specifically configured to:

determine whether an action triggered by a trigger request currently executed is within the action cycle; and

if the action triggered by the trigger request currently executed is not within the action cycle, call, according to the trigger request to be executed, the corresponding action execution interface to process the trigger event.

[0073] In a possible implementation, the calling module 704 is specifically configured to:

if the action triggered by the trigger request currently executed is within the action cycle, determine whether an action triggered by the trigger request to be executed and the action triggered by the trigger request currently executed are mutually exclusive; if the action triggered by the trigger request to be executed and the action triggered by the trigger request currently executed are not mutually exclusive, call, according to the trigger request to be executed, the corresponding action execution interface to process the trigger event; and if the action triggered by the trigger request to be executed and the action triggered by the trigger request currently executed are mutually exclusive,

after the trigger request currently executed is executed, call, according to the trigger request to be executed, the corresponding action execution interface to process the trigger event.

[0074] In a possible implementation, the trigger request carries an activation identifier, and the activation identifier is used to indicate whether the action triggered by the trigger request is an activation action.

[0075] The calling module 704 is specifically configured to:

determine whether the action triggered by the trigger request currently executed is the activation action according to the activation identifier in the trigger request currently executed; and

if the action triggered by the trigger request currently executed is the activation action, determine that the action triggered by the trigger request currently executed is within the action cycle.

[0076] In a possible implementation, the calling module 704 is specifically configured to:

obtain a type of the action triggered by the trigger request to be executed and a type of the action triggered by the trigger request currently executed; determine whether the type of the action triggered by the trigger request to be executed is consistent with the type of the action triggered by the trigger request currently executed; and

if the type of the action triggered by the trigger request to be executed is consistent with the type of the action triggered by the trigger request currently executed, determine that the action triggered by the trigger request to be executed and the action triggered by the trigger request currently executed are not mutually exclusive.

[0077] In a possible implementation, after the trigger signal collision detection is performed, and before obtaining, according to the preset sequence, the trigger request from the preset action queue as the trigger request to be executed, the obtaining module 703 is further configured to:

call a corresponding hover processing interface to process a hover event according to a node where a trigger signal collides with the UI in a previous frame and which is detected through the trigger signal collision detection and according to a node where the trigger signal collides with the UI in a current frame and which is detected through the trigger signal collision detection, and display a processed result of the hover processing interface on the UI.

In a possible implementation, the hover processing interface includes a hover entry interface, a hover stop interface, and a hover end interface; and the obtaining module 703 is specifically configured

to:

compare the node where the trigger signal collides with the UI in the previous frame and the node where the trigger signal collides with the UI in the current frame;
 according to a node, in a comparison result, which is not among nodes where the trigger signal collides with the UI in the previous frame, but is among nodes where the trigger signal collides with the UI in the current frame, call the hover entry interface to process the hover event;
 according to a node, in the comparison result, which is among the nodes where the trigger signal collides with the UI in the previous frame, and is among the nodes where the trigger signal collides with the UI in the current frame, call the hover stop interface to process the hover event;
 and
 according to a node, in the comparison result, which is among the nodes where the trigger signal collides with the UI in the previous frame, and is not among the nodes where the trigger signal collides with the UI in the current frame, call the hover end interface to process the hover event.

[0078] In a possible implementation, the calling module 704 is specifically configured to:

obtain a type of an action triggered by the trigger request to be executed;
 determine the corresponding action execution interface to be called according to the type of the action triggered by the trigger request to be executed; and
 call the corresponding action execution interface to process the trigger event. In a possible implementation, the target execution component interacts with a management system.

[0079] The receiving module 701 is specifically configured to:

receive the trigger request sent by the management system, where the trigger request is received by the management system from the target input device, and the trigger request is used to instruct the management system to set the state of the target execution component to be active, and set the state of each of the remaining execution components in the plurality of execution components except the target execution component to be inactive.

[0080] The apparatus provided by the embodiment of the present disclosure can be used to perform the technical solutions of the method embodiments shown in FIG. 2 to FIG. 4, and implementation principles and technical effects thereof are similar, and the details are not described herein again in the embodiment of the present

disclosure.

[0081] Optionally, FIG. 8 schematically provides a schematic diagram of a possible basic hardware architecture of the target execution component described in the present disclosure.

[0082] Referring to FIG. 8, the target execution component includes at least one processor 801 and a communication interface 803. Further optionally, a memory 802 and a bus 804 may also be included.

[0083] In the target execution component, the number of a processor 801 may be one or more, and FIG. 8 only illustrates one of the processors 801. Optionally, the processor 801 may be a central processing unit (central processing unit, CPU), a graphics processing unit (graphics processing unit, GPU), or a digital signal processor (digital signal processor, DSP). If the target execution component has multiple processors 801, the types of multiple processors 801 may be different or the same. Optionally, the multiple processors 801 of the target execution component may also be integrated into a multi-core processor.

[0084] The memory 802 stores computer instructions and data. The memory 802 may store computer instructions and data required to implement the interaction method provided by the present disclosure. For example, the memory 802 stores instructions for implementing the steps of the interaction method. The memory 802 may be any one or any combination of the following storage media: a non-volatile memory (e.g., a read-only memory (Read-Only Memory, ROM), a solid state disk (Solid State Disk SSD), a hard disk (Hard Disk Drive, HDD), an optical disc) and a volatile memory.

[0085] The communication interface 803 may provide information input/output for the at least one processor. It may also include any one or any combination of the following devices: a network interface (such as an Ethernet interface), a wireless network card, and other devices with network access function.

[0086] Optionally, the communication interface 803 may also be used for data communication between the target execution component and other computing devices or terminals.

[0087] Further optionally, FIG. 8 represents the bus 804 with a thick line. The bus 804 may connect the processor 801 with the memory 802 and the communication interface 803. In this way, through the bus 804, the processor 801 can access the memory 802, and can also use the communication interface 803 to perform data interaction with other computing devices or terminals.

[0088] In the present disclosure, the target execution component executes the computer instructions in the memory 802, so that the target execution component implements the interaction methods provided by the present disclosure, or deploys the interaction apparatus 70.

[0089] From the perspective of logical function division, for example, as shown in FIG. 8, the memory 802 may include a receiving module 701, a processing mod-

ule 702, an obtaining module 703 and a calling module 704. The inclusion here only refers to that the functions of the receiving module, the processing module, the obtaining module and the calling module can be implemented respectively when the instructions stored in the memory are executed, and it is not limited to be a physical structure.

[0090] The present disclosure provides a computer-readable storage medium, where the computer-readable program product includes computer instructions, and the computer instructions instruct a computing device to execute the interaction methods provided by the present disclosure.

[0091] The present disclosure provides a computer program product including computer instructions, and the computer instructions are executed by a processor to execute the interaction methods.

[0092] The present disclosure provides a computer program, where the computer program is stored in a readable storage medium, at least one processor of an electronic device can read the computer program from the readable storage medium, and the at least one processor executes the computer program, so that the electronic device executes the method provided by any one of the embodiments.

[0093] The present disclosure provides a chip including at least one processor and a communication interface, and the communication interface provides information input and/or output for the at least one processor. Further, the chip may also include at least one memory for storing computer instructions. The at least one processor is configured to call and execute the computer instructions to implement the interaction methods provided by the present disclosure.

[0094] In the several embodiments provided by the present disclosure, it should be understood that the disclosed apparatuses and methods may be implemented in other manners. For example, the apparatus embodiments described above are only illustrative. For example, the division of the units is only a logical function division. In the actual implementations, there may be other division manners. For example, multiple units or components may be combined or integrated into another system, or some features can be ignored, or not implemented. On the other hand, mutual coupling, direct coupling or communication connection shown or discussed may be indirect coupling or communication connection of apparatuses or units through some interfaces, and may be in electrical, mechanical or other forms.

[0095] The units described as separate components may or may not be separated in physical, and components displayed as units may or may not be physical units, that is, may be located in one place, or may be distributed to multiple network units. Some or all of the units may be selected according to actual needs to achieve the purpose of the solution in the embodiments.

[0096] In addition, each functional unit in each embodiment of the present disclosure may be integrated into one

processing unit, or each unit may exist alone in physical, or two or more units may be integrated into one unit. The integrated unit may be implemented in the form of hardware, or may be implemented in the form of hardware plus software functional units.

Claims

1. An interaction method, applied to a target execution component (101), and used for interacting with a virtual object in an extended reality scenario, wherein the target execution component (101) is one of a plurality of execution components (101), a state of the target execution component is active, and a state of each remaining execution component in the plurality of execution components except the target execution component (101) is inactive, and each execution component (101) corresponds to one input device, and the method comprises:

receiving (201) a trigger request sent by the corresponding target input device;
 caching (202) the trigger request in a preset action queue, and performing (202) trigger signal collision detection based on a user interface after receiving an update instruction;
 obtaining (203), according to a preset sequence, a trigger request from the preset action queue as a trigger request to be executed, after the trigger signal collision detection is performed; and
 calling (204), according to the trigger request to be executed, an action execution interface to process a trigger event, so that a processed result of the action execution interface is displayed on the user interface;
 wherein the received trigger signal causes the performing (202) of a trigger signal collision detection, wherein performing of the trigger signal collision detection comprises detecting the collision of a ray and the user interface after receiving an update instruction further comprises:
 tracking a position and an attitude of the corresponding target input device to determine a starting point and a direction of the ray, updating the position and the direction of the ray, and performing ray collision detection to determine a node where the ray collides with the user interface, and displaying the node, where a collision is occurred, on the user interface.

2. The method according to claim 1, before calling (204), according to the trigger request to be executed, the action execution interface to process the trigger event, further comprising:

determining whether an action triggered by a trigger request currently executed is within an action cycle; and
the calling (204), according to the trigger request to be executed, the action execution interface to process the trigger event comprises:
if the action triggered by the trigger request currently executed is not within the action cycle, calling, according to the trigger request to be executed, the action execution interface to process the trigger event.

3. The method according to claim 2, after determining whether the action triggered by the trigger request currently executed is within the action cycle, further comprising:

if the action triggered by the trigger request currently executed is within the action cycle, determining whether an action triggered by the trigger request to be executed and the action triggered by the trigger request currently executed are mutually exclusive;
the calling (204), according to the trigger request to be executed, the action execution interface to process the trigger event comprises:

if the action triggered by the trigger request to be executed and the action triggered by the trigger request currently executed are not mutually exclusive, calling, according to the trigger request to be executed, the corresponding action execution interface to process the trigger event; and
if the action triggered by the trigger request to be executed and the action triggered by the trigger request currently executed are mutually exclusive, after the trigger request currently executed is executed, calling, according to the trigger request to be executed, the action execution interface to process the trigger event.

4. The method according to claim 2 or 3, wherein the trigger request carries an activation identifier, and the activation identifier is used to indicate whether the action triggered by the trigger request is an activation action; and
the determining whether the action triggered by the trigger request currently executed is within the action cycle comprises:

determining whether the action triggered by the trigger request currently executed is the activation action according to the activation identifier in the trigger request currently executed; and
if the action triggered by the trigger request currently executed is the activation action, de-

termining that the action triggered by the trigger request currently executed is within the action cycle.

5. The method according to claim 3, wherein the determining whether the action triggered by the trigger request to be executed and the action triggered by the trigger request currently executed are mutually exclusive comprises:

obtaining a type of the action triggered by the trigger request to be executed and a type of the action triggered by the trigger request currently executed;
determining whether the type of the action triggered by the trigger request to be executed is consistent with the type of the action triggered by the trigger request currently executed; and
if the type of the action triggered by the trigger request to be executed is consistent with the type of the action triggered by the trigger request currently executed, determining that the action triggered by the trigger request to be executed and the action triggered by the trigger request currently executed are not mutually exclusive.

6. The method according to any one of claims 1 to 5, after the trigger signal collision detection is performed, and before obtaining, according to the preset sequence, the trigger request from the preset action queue as the trigger request to be executed, further comprising:

calling a hover processing interface to process a hover event according to a node where a trigger signal collides with the user interface in a previous frame and which is detected through the trigger signal collision detection and according to a node where the trigger signal collides with the user interface in a current frame and which is detected through the trigger signal collision detection, and displaying a processed result of the hover processing interface on the user interface.

7. The method according to claim 6, wherein the hover processing interface comprises a hover entry interface, a hover stop interface, and a hover end interface;

the calling the hover processing interface to process the hover event according to the node where the trigger signal collides with the user interface in the previous frame and which is detected through the trigger signal collision detection and according to the node where the trigger signal collides with the user interface in the current frame and which is detected through the trigger signal collision detection comprises:

comparing the node where the trigger signal

- collides with the user interface in the previous frame and the node where the trigger signal collides with the user interface in the current frame;
- according to a node, in a comparison result, which is not among nodes where the trigger signal collides with the user interface in the previous frame, but is among nodes where the trigger signal collides with the user interface in the current frame, calling the hover entry interface to process the hover event;
- according to a node, in the comparison result, which is among the nodes where the trigger signal collides with the user interface in the previous frame, and is among the nodes where the trigger signal collides with the user interface in the current frame, calling the hover stop interface to process the hover event; and
- according to a node, in the comparison result, which is among the nodes where the trigger signal collides with the user interface in the previous frame, but is not among the nodes where the trigger signal collides with the user interface in the current frame, calling the hover end interface to process the hover event.
8. The method according to any one of claims 1 to 3, wherein the calling, according to the trigger request to be executed, the action execution interface to process the trigger event comprises:
- obtaining a type of an action triggered by the trigger request to be executed;
- determining the action execution interface to be called according to the type of the action triggered by the trigger request to be executed; and
- calling the action execution interface to process the trigger event.
9. The method according to any one of claims 1 to 3, wherein the target execution component (101) interacts with a management system; and the receiving (201) the trigger request comprises: receiving the trigger request sent by the management system, wherein the trigger request is received by the management system from the target input device, and the trigger request is used to instruct the management system to set the state of the target execution component (101) to be active, and set the state of each of the remaining execution components (101) in the plurality of execution components (101) except the target execution component (101) to be inactive.
10. A target execution component (101), and adapted to be used for interacting with a virtual object in an extended reality scenario, wherein the target execution component (101) is one of a plurality of execution components (101), and each execution component (101) is configured to correspond to one input device, the target execution component comprises a communications interface (803) and an interaction apparatus (702), wherein the interaction apparatus (702) comprises:
- a receiving module (701), configured to receive a trigger request sent by a corresponding target input device in of a state where the target execution component (101) is active, and a state of each remaining execution component (101) in the plurality of execution components (101) except the target execution component (101) is inactive;
- a processing module (702), configured to cache the trigger request in a preset action queue, and perform trigger signal collision detection based on a user interface after receiving an update instruction;
- an obtaining module (703), configured to obtain, according to a preset sequence, a trigger request from the preset action queue as a trigger request to be executed, after the trigger signal collision detection is performed; and
- a calling module (704), configured to call, according to the trigger request to be executed, a corresponding action execution interface to process a trigger event, so that a processed result of the action execution interface is displayed on the user interface;
- wherein the received trigger signal causes the performing of a trigger signal collision detection and the processing module (702) is further configured to:
- track a position and an attitude of the corresponding target input device to determine a starting point and a direction of the ray, update the position and the direction of a ray, and perform ray collision detection to determine a node where the ray collides with the user interface, and display the node, where a collision is occurred, on the user interface.
11. The apparatus according to claim 10, wherein the calling module (704) is configured to:
- determine whether an action triggered by a trigger request currently executed is within an action cycle; and
- if the action triggered by the trigger request currently executed is not within the action cycle, call, according to the trigger request to be executed, the corresponding action execution interface to process the trigger event.
12. The apparatus according to claim 11, wherein the calling module (704) is configured to:

if the action triggered by the trigger request currently executed is within the action cycle, determine whether an action triggered by the trigger request to be executed and the action triggered by the trigger request currently executed are mutually exclusive; 5
if the action triggered by the trigger request to be executed and the action triggered by the trigger request currently executed are not mutually exclusive, call, according to the trigger request to be executed, the corresponding action execution interface to process the trigger event; and 10
if the action triggered by the trigger request to be executed and the action triggered by the trigger request currently executed are mutually exclusive, after the trigger request currently executed is executed, call, according to the trigger request to be executed, the action execution interface to process the trigger event. 15

13. A computer-readable storage medium, wherein the computer-readable storage medium stores a computer program, and the computer program causes a server to perform the method according to any one of claims 1 to 9. 20 25
14. A computer program product, comprising computer instructions to be executed by a processor to perform the method according to any one of claims 1 to 9. 30

Patentansprüche

1. Interaktionsverfahren, das auf eine Zielausführungskomponente (101) angewendet wird und zum Interagieren mit einem virtuellen Objekt in einem erweiterten Realitätsszenario verwendet wird, wobei die Zielausführungskomponente (101) eine von einer Vielzahl von Ausführungskomponenten (101) ist, ein Zustand der Zielausführungskomponente aktiv ist und ein Zustand jeder verbleibenden Ausführungskomponente in der Vielzahl von Ausführungskomponenten mit Ausnahme der Zielausführungskomponente inaktiv ist, und jede Ausführungskomponente (101) einer Eingabevorrichtung entspricht, und das Verfahren umfasst: 35 40 45

Empfangen (201) einer von der entsprechenden Zieleingabevorrichtung gesendeten Auslöseanforderung; 50
Zwischenspeichern (202) der Auslöseanforderung in einer voreingestellten Aktionswarteschlange und Durchführen (202) einer Auslösesignalkollisionserkennung basierend auf einer Benutzerschnittstelle nach Empfangen einer Aktualisierungsanweisung; 55
Erhalten (203), gemäß einer voreingestellten Sequenz, einer Auslöseanforderung aus der

voreingestellten Aktionswarteschlange als auszuführende Auslöseanforderung, nachdem die Auslösesignalkollisionserkennung durchgeführt worden ist; und
Aufrufen (204), gemäß der auszuführenden Auslöseanforderung, einer Aktionsausführungsschnittstelle, um ein Auslöseereignis zu verarbeiten, sodass ein verarbeitetes Ergebnis der Aktionsausführungsschnittstelle auf der Benutzeroberfläche angezeigt wird;
wobei das empfangene Auslösesignal das Durchführen (202) einer Auslösesignalkollisionserkennung veranlasst, wobei Durchführen der Auslösesignalkollisionserkennung Erkennen der Kollision eines Strahls umfasst, und die Benutzeroberfläche nach dem Empfang einer Aktualisierungsanweisung weiter umfasst: Verfolgen einer Position und einer Haltung der entsprechenden Zieleingabevorrichtung, um einen Startpunkt und eine Richtung des Strahls zu bestimmen, Aktualisieren der Position und der Richtung des Strahls und Durchführen einer Strahlkollisionserkennung, um einen Knoten zu bestimmen, wo der Strahl mit der Benutzeroberfläche kollidiert, und Anzeigen des Knotens, wo eine Kollision aufgetreten ist, auf der Benutzeroberfläche.

2. Verfahren nach Anspruch 1, das vor Aufrufen (204), gemäß der auszuführenden Auslöseanforderung, der Aktionsausführungsschnittstelle, um ein Auslöseereignis zu verarbeiten, weiter umfasst: 30

Bestimmen, ob eine durch eine aktuell ausgeführte Auslöseanforderung ausgelöste Aktion innerhalb eines Aktionszyklus liegt; und
das Aufrufen (204), gemäß der auszuführenden Auslöseanforderung, der Aktionsausführungsschnittstelle, um das Auslöseereignis zu verarbeiten, umfasst:
falls die durch die aktuell ausgeführte Auslöseanforderung ausgelöste Aktion nicht innerhalb des Aktionszyklus liegt, Aufrufen, gemäß der auszuführenden Auslöseanforderung, der Aktionsausführungsschnittstelle, um das Auslöseereignis zu verarbeiten.

3. Verfahren nach Anspruch 2, das nach Bestimmen, ob die durch die aktuell ausgeführte Auslöseanforderung ausgelöste Aktion innerhalb des Aktionszyklus liegt, weiter umfasst: 50

falls die durch die aktuell ausgeführte Auslöseanforderung ausgelöste Aktion innerhalb des Aktionszyklus liegt, Bestimmen, ob sich eine durch die auszuführende Auslöseanforderung ausgelöste Aktion und die durch die aktuell ausgeführte Auslöseanforderung ausgelöste

Aktion gegenseitig ausschließen;
das Aufrufen (204), gemäß der auszuführenden
Auslöseanforderung, der Aktionsausführungs-
schnittstelle, um das Auslöseereignis zu verar-
beiten, umfasst:

falls sich die durch die auszuführende Aus-
löseanforderung ausgelöste Aktion und die
durch die aktuell ausgeführte Auslösean-
forderung ausgelöste Aktion nicht gegen-
seitig ausschließen, Aufrufen, gemäß der
auszuführenden Auslöseanforderung, der
entsprechenden Aktionsausführungs-
schnittstelle, um das Auslöseereignis zu
verarbeiten; und

falls sich die durch die auszuführende Aus-
löseanforderung ausgelöste Aktion und die
durch die aktuell ausgeführte Auslösean-
forderung ausgelöste Aktion gegenseitig
ausschließen, nachdem die aktuell ausge-
führte Auslöseanforderung ausgeführt wor-
den ist, Aufrufen, gemäß der auszuführen-
den Auslöseanforderung, der Aktionsaus-
führungsschnittstelle, um das Auslöseer-
eignis zu verarbeiten.

4. Verfahren nach Anspruch 2 oder 3, wobei die Aus-
löseanforderung eine Aktivierungskennung trägt
und die Aktivierungskennung verwendet wird, um
anzugeben, ob die durch die Auslöseanforderung
ausgelöste Aktion eine Aktivierungsaktion ist; und
das Bestimmen, ob die durch die aktuell ausgeführte
Auslöseanforderung ausgelöste Aktion innerhalb
des Aktionszyklus liegt, umfasst:

Bestimmen, ob die durch die aktuell ausgeführte
Auslöseanforderung ausgelöste Aktion die Ak-
tivierungsaktion gemäß der Aktivierungsken-
nung in der aktuell ausgeführten Auslöseanfor-
derung ist; und
wenn die durch die aktuell ausgeführte Auslö-
seanforderung ausgelöste Aktion die Aktivi-
erungsaktion ist, Bestimmen, dass die durch
die aktuell ausgeführte Auslöseanforderung
ausgelöste Aktion innerhalb des Aktionszyklus
liegt.

5. Verfahren nach Anspruch 3, wobei das Bestimmen,
ob sich die durch die auszuführende Auslöseanfor-
derung ausgelöste Aktion und die durch die aktuell
ausgeführte Auslöseanforderung ausgelöste Aktion
gegenseitig ausschließen, umfasst:

Erhalten eines Typs der Aktion, die durch die
auszuführende Auslöseanforderung ausgelöst
wird, und eines Typs der Aktion, die durch die
aktuell ausgeführte Auslöseanforderung ausge-
löst wird;

Bestimmen, ob der Typ der Aktion, die durch die
auszuführende Auslöseanforderung ausgelöst
wird, mit dem Typ der Aktion, die durch die
aktuell ausgeführte Auslöseanforderung ausge-
löst wird, übereinstimmt; und
wenn der Typ der durch die auszuführende Aus-
löseanforderung ausgelösten Aktion mit dem
Typ der durch die aktuell ausgeführte Auslö-
seanforderung ausgelösten Aktion überein-
stimmt, Bestimmen, dass sich die durch die
auszuführende Auslöseanforderung ausgelös-
te Aktion und die durch die aktuell ausgeführte
Auslöseanforderung ausgelöste Aktion nicht
gegenseitig ausschließen.

6. Verfahren nach einem der Ansprüche 1 bis 5, das,
nachdem die Auslösesignalkollisionserkennung
durchgeführt worden ist, und vor Erhalten, gemäß
der voreingestellten Sequenz, der Auslöseanfor-
derung aus der voreingestellten Aktionswarteschlange
als auszuführende Auslöseanforderung weiter um-
fasst:

Aufrufen einer Hover-Verarbeitungsschnittstelle, um
ein Hover-Ereignis gemäß einem Knoten, wo ein
Auslösesignal mit der Benutzeroberfläche in einem
vorherigen Frame kollidiert und der durch die Aus-
lösesignalkollisionserkennung erkannt worden ist,
und gemäß einem Knoten, wo das Auslösesignal
mit der Benutzeroberfläche in einem aktuellen
Frame kollidiert und der durch die Auslösesignalkol-
lisionserkennung erkannt worden ist, zu verarbeiten,
und Anzeigen eines verarbeiteten Ergebnisses der
Hover-Verarbeitungsschnittstelle auf der Benutze-
roberfläche.

7. Verfahren nach Anspruch 6, wobei die Hover-Ver-
arbeitungsschnittstelle eine Hover-Eintrittsschnitt-
stelle, eine Hover-Stopp-Schnittstelle und eine Ho-
ver-End-Schnittstelle umfasst;
das Aufrufen der Hover-Verarbeitungsschnittstelle,
um das Hover-Ereignis gemäß dem Knoten, wo das
Auslösesignal mit der Benutzeroberfläche im vor-
herigen Frame kollidiert und der durch die Auslöse-
signalkollisionserkennung erkannt worden ist, und
gemäß dem Knoten, wo das Auslösesignal mit der
Benutzeroberfläche im aktuellen Frame kollidiert
und der durch die Auslösesignalkollisionserkennung
erkannt worden ist, zu verarbeiten, umfasst:

Vergleichen des Knotens, an dem das Auslöse-
signal mit der Benutzeroberfläche im vorherigen
Frame kollidiert, und des Knotens, wo das Aus-
lösesignal mit der Benutzeroberfläche im ak-
tuellen Frame kollidiert;
gemäß einem Knoten, in einem Vergleichser-
gebnis, der sich nicht unter den Knoten befindet,
wo das Auslösesignal mit der Benutzeroberflä-
che im vorherigen Frame kollidiert, sondern un-

- ter den Knoten, wo das Auslösesignal mit der Benutzeroberfläche im aktuellen Frame kollidiert, Aufrufen der Hover-Eintrittsschnittstelle, um das Hover-Ereignis zu verarbeiten; gemäß einem Knoten, in dem Vergleichsergebnis, der sich unter den Knoten befindet, wo das Auslösesignal mit der Benutzeroberfläche im vorherigen Frame kollidiert, und unter den Knoten, wo das Auslösesignal mit der Benutzeroberfläche im aktuellen Frame kollidiert, Aufrufen der Hover-Stopp-Schnittstelle, um das Hover-Ereignis zu verarbeiten; und gemäß einem Knoten, in dem Vergleichsergebnis, der sich unter den Knoten befindet, wo das Auslösesignal mit der Benutzeroberfläche im vorherigen Frame kollidiert, jedoch nicht unter den Knoten, wo das Auslösesignal mit der Benutzeroberfläche im aktuellen Frame kollidiert, Aufrufen der Hover-End-Schnittstelle, um das Hover-Ereignis zu verarbeiten.
8. Verfahren nach einem der Ansprüche 1 bis 3, wobei das Aufrufen, gemäß der auszuführenden Auslöseanforderung, der Aktionsausführungsschnittstelle, um das Auslöseereignis zu verarbeiten, umfasst:
- Erhalten eines Typs einer Aktion, die durch die auszuführende Auslöseanforderung ausgelöst wird;
 - Bestimmen der aufzurufenden Aktionsausführungsschnittstelle gemäß dem Typ der Aktion, die durch die auszuführende Auslöseanforderung ausgelöst wird; und
 - Aufrufen der Aktionsausführungsschnittstelle, um das Auslöseereignis zu verarbeiten.
9. Verfahren nach einem der Ansprüche 1 bis 3, wobei die Zielausführungskomponente (101) mit einem Verwaltungssystem interagiert; und das Empfangen (201) der Auslöseanforderung umfasst:
- Empfangen der von dem Verwaltungssystem gesendeten Auslöseanforderung, wobei die Auslöseanforderung von dem Verwaltungssystem von der Zieleingabevorrichtung empfangen wird und die Auslöseanforderung verwendet wird, um das Verwaltungssystem anzuweisen, den Zustand der Zielausführungskomponente (101) auf aktiv zu setzen und den Zustand jeder der verbleibenden Ausführungskomponenten (101) in der Vielzahl von Ausführungskomponenten (101) mit Ausnahme der Zielausführungskomponente (101) auf inaktiv zu setzen.
10. Zielausführungskomponente (101), und die zum Interagieren mit einem virtuellen Objekt in einem erweiterten Realitätsszenario angepasst ist, wobei die Zielausführungskomponente (101) eine von einer

Vielzahl von Ausführungskomponenten (101) ist und jede Ausführungskomponente (101) konfiguriert ist, um einer Eingabevorrichtung zu entsprechen, die Zielausführungskomponente eine Kommunikationsschnittstelle (803) und eine Interaktionseinrichtung (702) umfasst, wobei die Interaktionseinrichtung (702) umfasst:

ein Empfangsmodul (701), das konfiguriert ist, um eine von einer entsprechenden Zieleingabevorrichtung gesendete Auslöseanforderung in einem Zustand zu empfangen, in dem die Zielausführungskomponente (101) aktiv ist, und ein Zustand jeder verbleibenden Ausführungskomponente (101) in der Vielzahl von Ausführungskomponenten (101) mit Ausnahme der inaktiven Zielausführungskomponente (101) inaktiv ist; ein Verarbeitungsmodul (702), das konfiguriert ist, um die Auslöseanforderung in einer voreingestellten Aktionswarteschlange zwischenspeichern, und nach Empfangen einer Aktualisierungsanweisung eine Auslösesignalkollisionserkennung basierend auf einer Benutzeroberfläche durchzuführen; ein Erhaltungsmodul (703), das konfiguriert ist, um, gemäß einer voreingestellten Sequenz, eine Auslöseanforderung aus der voreingestellten Aktionswarteschlange als eine auszuführende Auslöseanforderung zu erhalten, nachdem die Auslösesignalkollisionserkennung durchgeführt worden ist; und ein Aufrufmodul (704), das konfiguriert ist, um, gemäß der auszuführenden Auslöseanforderung, eine entsprechende Aktionsausführungsschnittstelle aufzurufen, um ein Auslöseereignis zu verarbeiten, sodass ein verarbeitetes Ergebnis der Aktionsausführungsschnittstelle auf der Benutzeroberfläche angezeigt wird; wobei das empfangene Auslösesignal das Durchführen einer Auslösesignalkollisionserkennung veranlasst und das Verarbeitungsmodul (702) weiter konfiguriert ist, um: eine Position und eine Einstellung der entsprechenden Zieleingabevorrichtung zu verfolgen, um einen Startpunkt und eine Richtung des Strahls zu bestimmen, die Position und die Richtung des Strahls zu aktualisieren, und Strahlkollisionserkennung durchzuführen, um einen Knoten zu bestimmen, wo der Strahl mit der Benutzeroberfläche kollidiert, und den Knoten, wo eine Kollision aufgetreten ist, auf der Benutzeroberfläche anzuzeigen.

11. Einrichtung nach Anspruch 10, wobei das Aufrufmodul (704) konfiguriert ist, um:

zu bestimmen, ob eine durch eine aktuell ausgeführte Auslöseanforderung ausgelöste Ak-

tion innerhalb eines Aktionszyklus liegt; und falls die durch die aktuell ausgeführte Auslöseanforderung ausgelöste Aktion nicht innerhalb des Aktionszyklus liegt, gemäß der auszuführenden Auslöseanforderung die Aktionsausführungsschnittstelle aufzurufen, um das Auslöseereignis zu verarbeiten.

12. Einrichtung nach Anspruch 11, wobei das Aufrufmodul (704) konfiguriert ist, um:

falls die durch die aktuell ausgeführte Auslöseanforderung ausgelöste Aktion innerhalb des Aktionszyklus liegt, zu bestimmen, ob sich eine durch die auszuführende Auslöseanforderung ausgelöste Aktion und die durch die aktuell ausgeführte Auslöseanforderung ausgelöste Aktion gegenseitig ausschließen; falls sich die durch die auszuführende Auslöseanforderung ausgelöste Aktion und die durch die aktuell ausgeführte Auslöseanforderung ausgelöste Aktion nicht gegenseitig ausschließen, gemäß der auszuführenden Auslöseanforderung die entsprechende Aktionsausführungsschnittstelle aufzurufen, um das Auslöseereignis zu verarbeiten; und falls sich die durch die auszuführende Auslöseanforderung ausgelöste Aktion und die durch die aktuell ausgeführte Auslöseanforderung ausgelöste Aktion gegenseitig ausschließen, nachdem die aktuell ausgeführte Auslöseanforderung ausgeführt worden ist, gemäß der auszuführenden Auslöseanforderung die Aktionsausführungsschnittstelle aufzurufen, um das Auslöseereignis zu verarbeiten.

13. Computerlesbares Speichermedium, wobei das computerlesbare Speichermedium ein Computerprogramm speichert und das Computerprogramm einen Server veranlasst, das Verfahren nach einem der Ansprüche 1 bis 9 durchzuführen.

14. Computerprogrammprodukt, das Computeranweisungen umfasst, die von einem Prozessor auszuführen sind, um das Verfahren nach einem der Ansprüche 1 bis 9 durchzuführen.

Revendications

1. Procédé d'interaction, appliqué à un composant d'exécution cible (101) et utilisé pour interagir avec un objet virtuel dans un scénario de réalité étendue, dans lequel le composant d'exécution cible (101) est un composant d'exécution d'une pluralité de composants d'exécution (101), un état du composant d'exécution cible est actif, et un état de chaque composant d'exécution restant dans la pluralité de

composants d'exécution, à l'exception du composant d'exécution cible, est inactif, et chaque composant d'exécution (101) correspond à un dispositif d'entrée, et le procédé comprend :

la réception (201) d'une demande de déclenchement envoyée par le dispositif d'entrée cible correspondant ;

la mise en cache (202) de la demande de déclenchement dans une file d'attente d'actions prédéfinie et la réalisation (202) d'une détection de collision de signal de déclenchement sur la base d'une interface utilisateur après réception d'une instruction de mise à jour ;

l'obtention (203), selon une séquence prédéfinie, d'une demande de déclenchement à partir de la file d'attente d'actions prédéfinie en tant que demande de déclenchement à exécuter, après que la détection de collision de signal de déclenchement est réalisée ; et

l'appel (204), selon la demande de déclenchement à exécuter, d'une interface d'exécution d'action pour traiter un événement de déclenchement, de telle sorte qu'un résultat traité de l'interface d'exécution d'action soit affiché sur l'interface utilisateur ;

dans lequel le signal de déclenchement reçu provoque la réalisation (202) d'une détection de collision de signal de déclenchement, dans lequel la réalisation de la détection de collision de signal de déclenchement comprend la détection de la collision d'un rayon et de l'interface utilisateur après réception d'une instruction de mise à jour comprend en outre :

le suivi d'une position et d'une attitude du dispositif d'entrée cible correspondant pour déterminer un point de départ et une direction du rayon, la mise à jour de la position et de la direction du rayon, et la réalisation d'une détection de collision de rayon pour déterminer un nœud où le rayon entre en collision avec l'interface utilisateur, et l'affichage du nœud, où une collision s'est produite, sur l'interface utilisateur.

2. Procédé selon la revendication 1, avant l'appel (204), selon la demande de déclenchement à exécuter, de l'interface d'exécution d'action pour traiter l'événement de déclenchement, comprenant en outre :

la détermination si une action déclenchée par une demande de déclenchement en cours d'exécution se trouve ou non dans un cycle d'action ; et

l'appel (204), selon la demande de déclenchement à exécuter, de l'interface d'exécution d'action pour traiter l'événement de déclenchement comprend :

si l'action déclenchée par la demande de déclenchement en cours d'exécution ne se trouve pas dans le cycle d'action, l'appel, selon la demande de déclenchement à exécuter, de l'interface d'exécution d'action pour traiter l'événement de déclenchement. 5

3. Procédé selon la revendication 2, après la détermination si l'action déclenchée par la demande de déclenchement en cours d'exécution se trouve dans le cycle d'action, comprenant en outre : 10

si l'action déclenchée par la demande de déclenchement en cours d'exécution se trouve dans le cycle d'action, la détermination si une action déclenchée par la demande de déclenchement à exécuter et l'action déclenchée par la demande de déclenchement en cours d'exécution s'excluent ou non mutuellement ; l'appel (204), selon la demande de déclenchement à exécuter, de l'interface d'exécution d'action pour traiter l'événement de déclenchement comprend : 20

si l'action déclenchée par la demande de déclenchement à exécuter et l'action déclenchée par la demande de déclenchement en cours d'exécution ne s'excluent pas mutuellement, l'appel, selon la demande de déclenchement à exécuter, de l'interface d'exécution d'action correspondante pour traiter l'événement de déclenchement ; et 25

si l'action déclenchée par la demande de déclenchement à exécuter et l'action déclenchée par la demande de déclenchement en cours d'exécution s'excluent mutuellement, après que la demande de déclenchement en cours d'exécution est exécutée, l'appel, selon la demande de déclenchement à exécuter, de l'interface d'exécution d'action pour traiter l'événement de déclenchement. 30

4. Procédé selon la revendication 2 ou 3, dans lequel la demande de déclenchement porte un identifiant d'activation et l'identifiant d'activation est utilisé pour indiquer si l'action déclenchée par la demande de déclenchement est ou non une action d'activation ; et la détermination si l'action déclenchée par la demande de déclenchement en cours d'exécution se trouve ou non dans le cycle d'action comprend : 45

la détermination si l'action déclenchée par la demande de déclenchement en cours d'exécution est ou non l'action d'activation selon l'identifiant d'activation dans la demande de déclenchement en cours d'exécution ; et 50

si l'action déclenchée par la demande de déclenchement en cours d'exécution est l'action d'activation, la détermination que l'action déclenchée par la demande de déclenchement en cours d'exécution se trouve dans le cycle d'action.

5. Procédé selon la revendication 3, dans lequel la détermination si l'action déclenchée par la demande de déclenchement à exécuter et l'action déclenchée par la demande de déclenchement en cours d'exécution s'excluent mutuellement comprend : 55

l'obtention d'un type de l'action déclenchée par la demande de déclenchement à exécuter et d'un type de l'action déclenchée par la demande de déclenchement en cours d'exécution ; la détermination si le type de l'action déclenchée par la demande de déclenchement à exécuter est ou non compatible avec le type de l'action déclenchée par la demande de déclenchement en cours d'exécution ; et si le type de l'action déclenchée par la demande de déclenchement à exécuter est compatible avec le type de l'action déclenchée par la demande de déclenchement en cours d'exécution, la détermination que l'action déclenchée par la demande de déclenchement à exécuter et l'action déclenchée par la demande de déclenchement en cours d'exécution ne sont pas mutuellement exclusives. 60

6. Procédé selon l'une quelconque des revendications 1 à 5, après la réalisation de la détection de collision de signal de déclenchement et avant l'obtention, selon la séquence prédéfinie, de la demande de déclenchement de la file d'attente d'actions prédéfinie en tant que demande de déclenchement à exécuter, comprenant en outre : 65

l'appel d'une interface de traitement de survol pour traiter un événement de survol selon un nœud où un signal de déclenchement entre en collision avec l'interface utilisateur dans une précédente trame et qui est détecté au moyen de la détection de collision de signal de déclenchement et selon un nœud où le signal de déclenchement entre en collision avec l'interface utilisateur dans une trame actuelle et qui est détecté au moyen de la détection de collision de signal de déclenchement, et l'affichage d'un résultat traité de l'interface de traitement de survol sur l'interface utilisateur. 70

7. Procédé selon la revendication 6, dans lequel l'interface de traitement de survol comprend une interface d'entrée de survol, une interface d'arrêt de survol et une interface de fin de survol ; 75

l'appel de l'interface de traitement de survol pour traiter l'événement de survol selon le nœud où le signal de déclenchement entre en collision avec l'interface utilisateur dans la précédente trame et qui est détecté au moyen de la détection de collision de signal de déclenchement et selon le nœud où le signal de déclenchement entre en collision avec l'interface utilisateur dans la trame actuelle et qui est détecté au moyen de la détection de collision de signal de déclenchement comprend :

la comparaison du nœud où le signal de déclenchement entre en collision avec l'interface utilisateur dans la précédente trame et du nœud où le signal de déclenchement entre en collision avec l'interface utilisateur dans la trame actuelle ;

selon un nœud, dans un résultat de comparaison, qui ne se trouve pas parmi les nœuds où le signal de déclenchement entre en collision avec l'interface utilisateur dans la précédente trame, mais se trouve parmi des nœuds où le signal de déclenchement entre en collision avec l'interface utilisateur dans la trame actuelle, l'appel de l'interface d'entrée de survol pour traiter l'événement de survol ;

selon un nœud, dans le résultat de comparaison, qui se trouve parmi les nœuds où le signal de déclenchement entre en collision avec l'interface utilisateur dans la précédente trame, et se trouve parmi les nœuds où le signal de déclenchement entre en collision avec l'interface utilisateur dans la trame actuelle, l'appel de l'interface d'arrêt de survol pour traiter l'événement de survol ; et

selon un nœud, dans le résultat de comparaison, qui se trouve parmi les nœuds où le signal de déclenchement entre en collision avec l'interface utilisateur dans la précédente trame, mais ne se trouve pas parmi les nœuds où le signal de déclenchement entre en collision avec l'interface utilisateur dans la trame actuelle, l'appel de l'interface de fin de survol pour traiter l'événement de survol.

8. Procédé selon l'une quelconque des revendications 1 à 3, dans lequel l'appel, selon la demande de déclenchement à exécuter, de l'interface d'exécution d'action pour traiter l'événement de déclenchement comprend :

l'obtention d'un type d'une action déclenchée par la demande de déclenchement à exécuter ; la détermination de l'interface d'exécution d'action à appeler selon le type de l'action déclenchée par la demande de déclenchement à exécuter ; et l'appel de l'interface d'exécution d'action pour

traiter l'événement de déclenchement.

9. Procédé selon l'une quelconque des revendications 1 à 3, dans lequel le composant d'exécution cible (101) interagit avec un système de gestion ; et la réception (201) de la demande de déclenchement comprend :

la réception de la demande de déclenchement envoyée par le système de gestion, dans lequel la demande de déclenchement est reçue par le système de gestion en provenance du dispositif d'entrée cible, et la demande de déclenchement est utilisée pour ordonner au système de gestion de définir l'état du composant d'exécution cible (101) comme étant actif, et de définir l'état de chacun des composants d'exécution (101) restants dans la pluralité de composants d'exécution (101) à l'exception du composant d'exécution cible (101) comme étant inactif.

10. Composant d'exécution cible (101) adapté pour être utilisé pour interagir avec un objet virtuel dans un scénario de réalité étendue, dans lequel le composant d'exécution cible (101) est un composant d'exécution d'une pluralité de composants d'exécution (101) et chaque composant d'exécution (101) est configuré pour correspondre à un dispositif d'entrée, le composant d'exécution cible comprenant une interface de communication (803) et un appareil d'interaction (702), dans lequel l'appareil d'interaction (702) comprend :

un module de réception (701), configuré pour recevoir une demande de déclenchement envoyée par un dispositif d'entrée cible correspondant dans un état dans lequel le composant d'exécution cible (101) est actif, et un état de chaque composant d'exécution (101) restant dans la pluralité de composants d'exécution (101) à l'exception du composant d'exécution cible (101) est inactif ;

un module de traitement (702), configuré pour mettre en cache la demande de déclenchement dans une file d'attente d'actions prédéfinie, et réaliser une détection de collision de signal de déclenchement sur la base d'une interface utilisateur après réception d'une instruction de mise à jour ;

un module d'obtention (703), configuré pour obtenir, selon une séquence prédéfinie, une demande de déclenchement à partir de la file d'attente d'actions prédéfinie en tant que demande de déclenchement à exécuter, après que la détection de collision de signal de déclenchement est réalisée ; et

un module d'appel (704), configuré pour appeler, selon la demande de déclenchement à exécuter, une interface d'exécution d'action corres-

pondante pour traiter un événement de déclenchement, de telle sorte qu'un résultat traité de l'interface d'exécution d'action soit affiché sur l'interface utilisateur ;
 dans lequel le signal de déclenchement reçu 5
 provoque l'exécution d'une détection de collision de signal de déclenchement et le module de traitement (702) est en outre configuré pour :
 suivre une position et une attitude du dispositif d'entrée cible correspondant pour déterminer 10
 un point de départ et une direction du rayon, mettre à jour la position et la direction d'un rayon, et réaliser une détection de collision de rayon pour déterminer un nœud où le rayon entre en collision avec l'interface utilisateur, et afficher le 15
 nœud où une collision s'est produite, sur l'interface utilisateur.

11. Appareil selon la revendication 10, dans lequel le module d'appel (704) est configuré pour : 20

déterminer si une action déclenchée par une demande de déclenchement en cours d'exécution se trouve ou non dans un cycle d'action ; et
 si l'action déclenchée par la demande de déclenchement en cours d'exécution ne se trouve 25
 pas dans le cycle d'action, appeler, selon la demande de déclenchement à exécuter, l'interface d'exécution d'action correspondante pour traiter l'événement de déclenchement. 30

12. Appareil selon la revendication 11, dans lequel le module d'appel (704) est configuré pour :

si l'action déclenchée par la demande de déclenchement en cours d'exécution se trouve 35
 dans le cycle d'action, déterminer si une action déclenchée par la demande de déclenchement à exécuter et l'action déclenchée par la demande de déclenchement en cours d'exécution 40
 s'excluent ou non mutuellement ;
 si l'action déclenchée par la demande de déclenchement à exécuter et l'action déclenchée par la demande de déclenchement en cours d'exécution ne s'excluent pas mutuellement, 45
 appeler, selon la demande de déclenchement à exécuter, l'interface d'exécution d'action correspondante pour traiter l'événement de déclenchement ; et
 si l'action déclenchée par la demande de déclenchement à exécuter et l'action déclenchée par la demande de déclenchement en cours d'exécution s'excluent mutuellement, après l'exécution de la demande de déclenchement en 50
 cours d'exécution, appeler, selon la demande de déclenchement à exécuter, l'interface d'exécution d'action pour traiter l'événement de déclenchement. 55

13. Support de stockage lisible par ordinateur, dans lequel le support de stockage lisible par ordinateur stocke un programme informatique, et le programme informatique amène un serveur à réaliser le procédé selon l'une quelconque des revendications 1 à 9.

14. Produit programme informatique comprenant des instructions informatiques à exécuter par un processeur pour réaliser le procédé selon l'une quelconque des revendications 1 à 9.

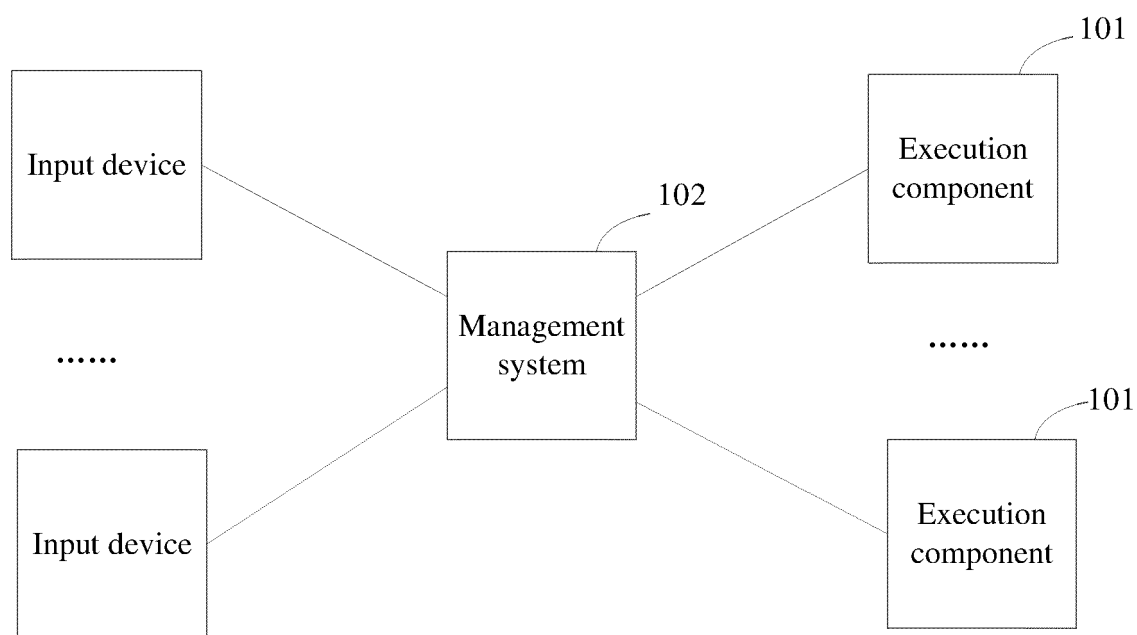


FIG. 1

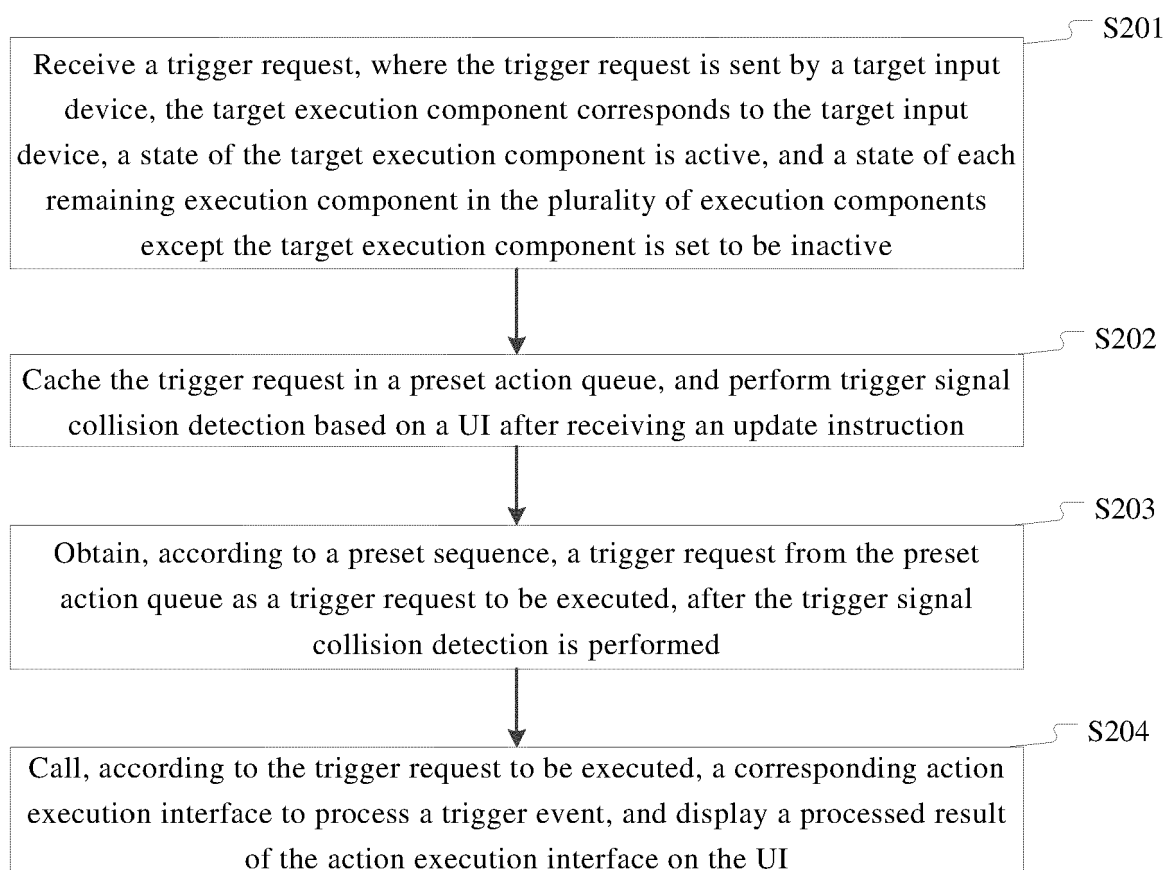


FIG. 2

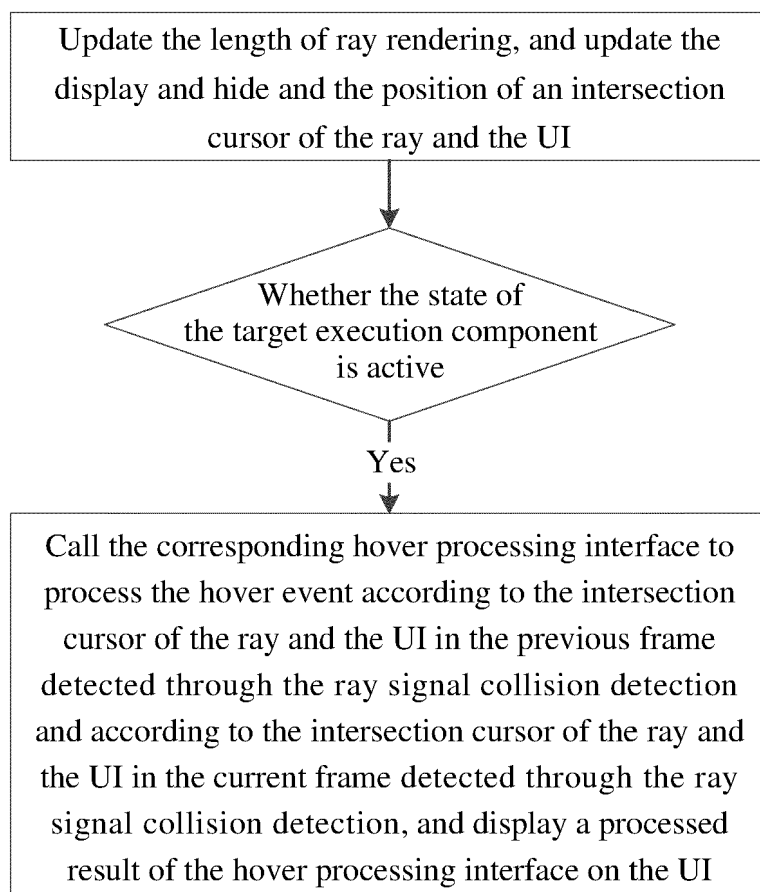


FIG. 3

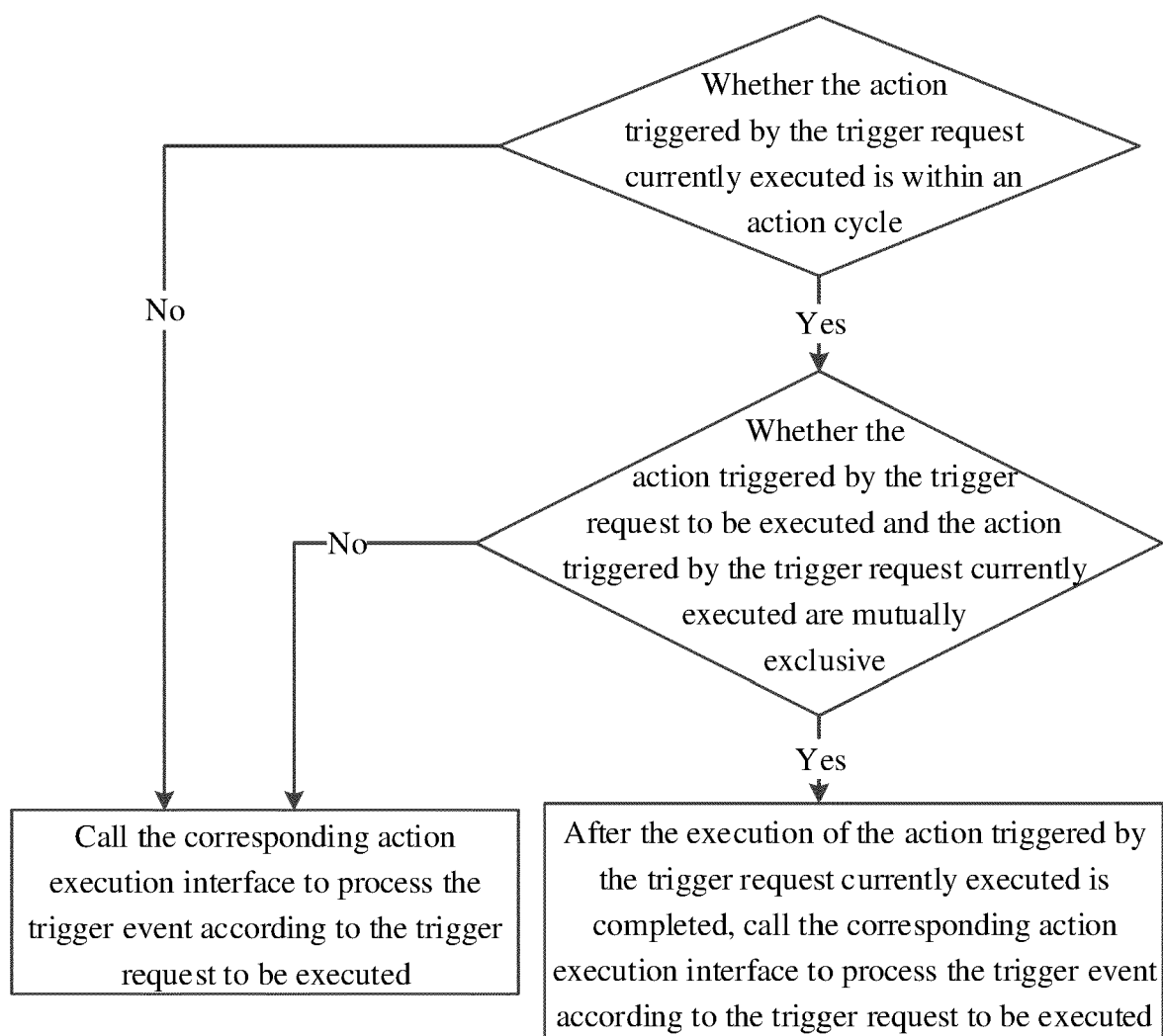


FIG. 4

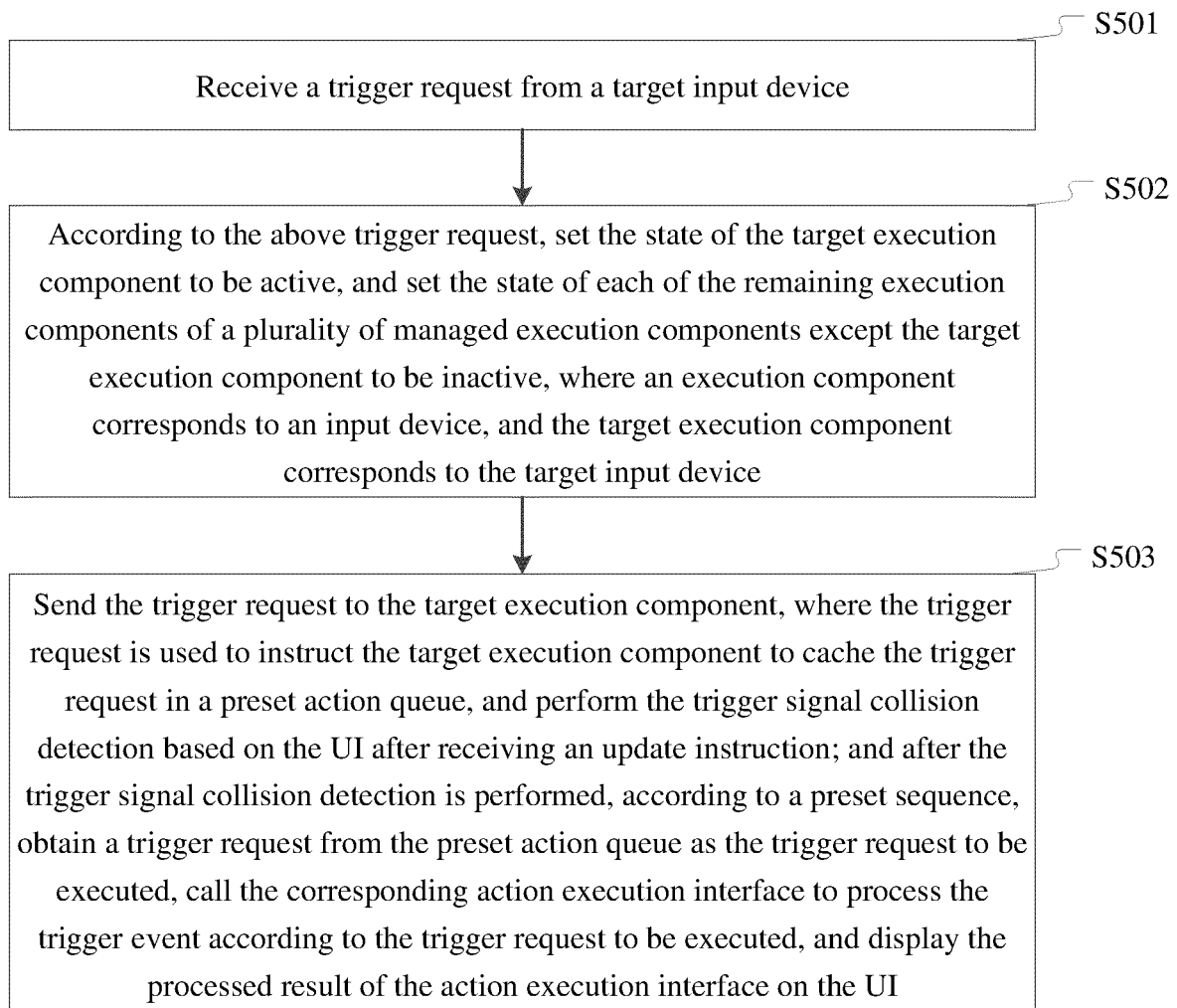


FIG. 5

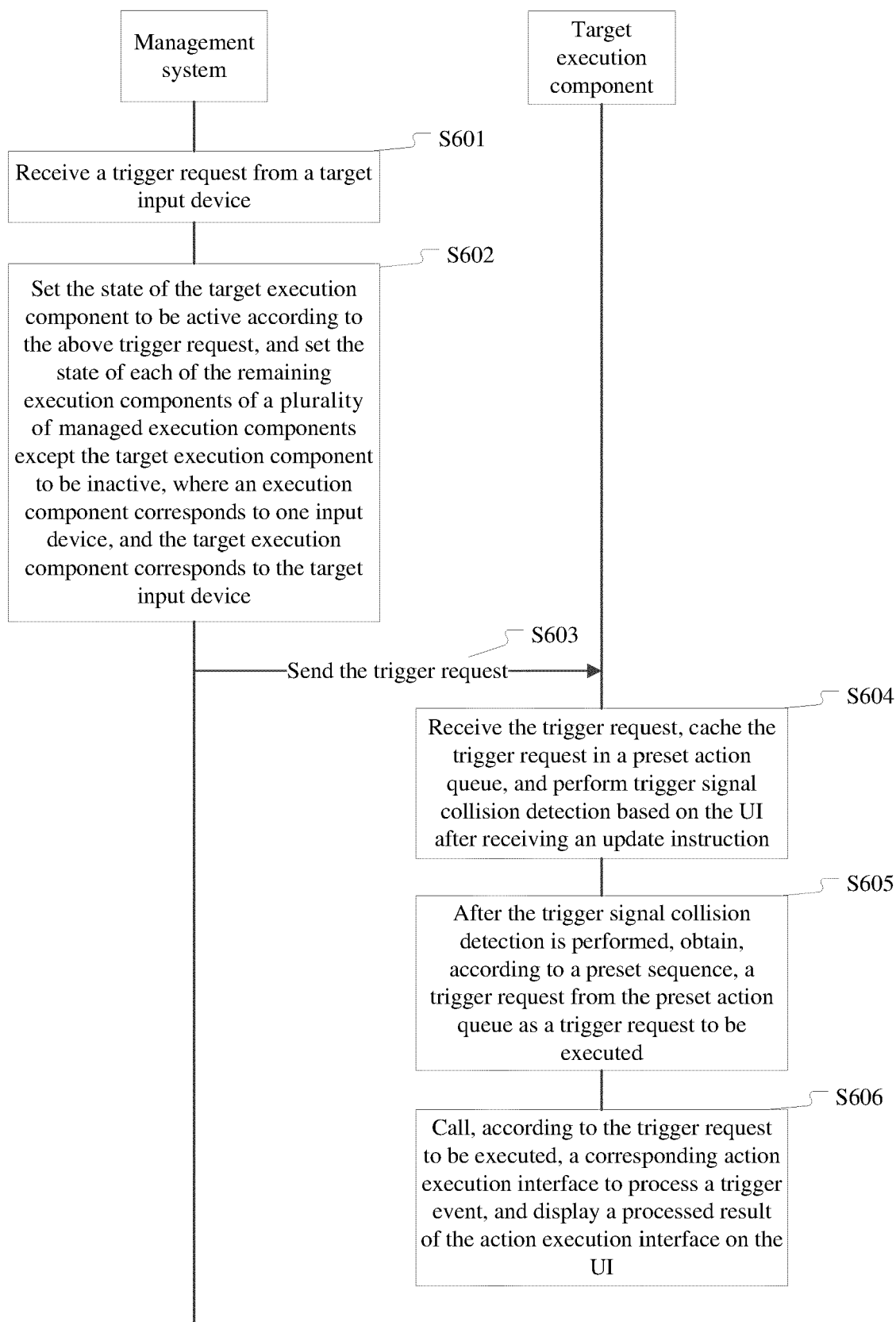


FIG. 6

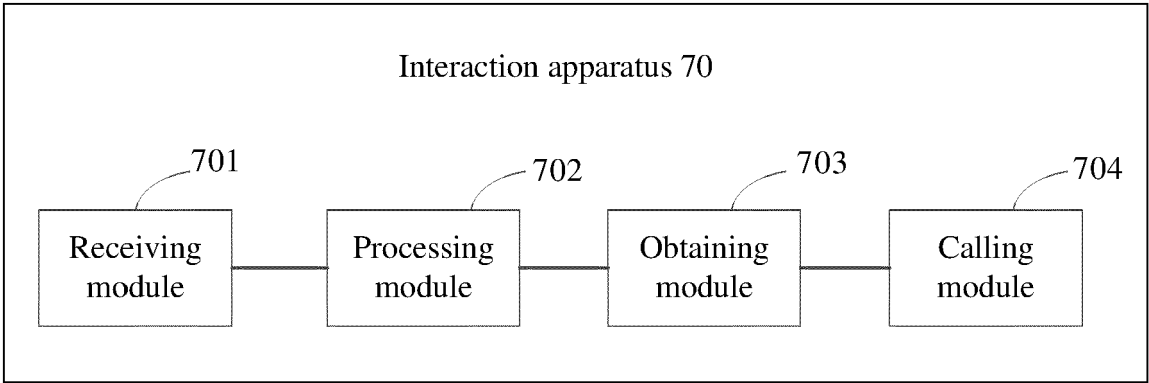


FIG. 7

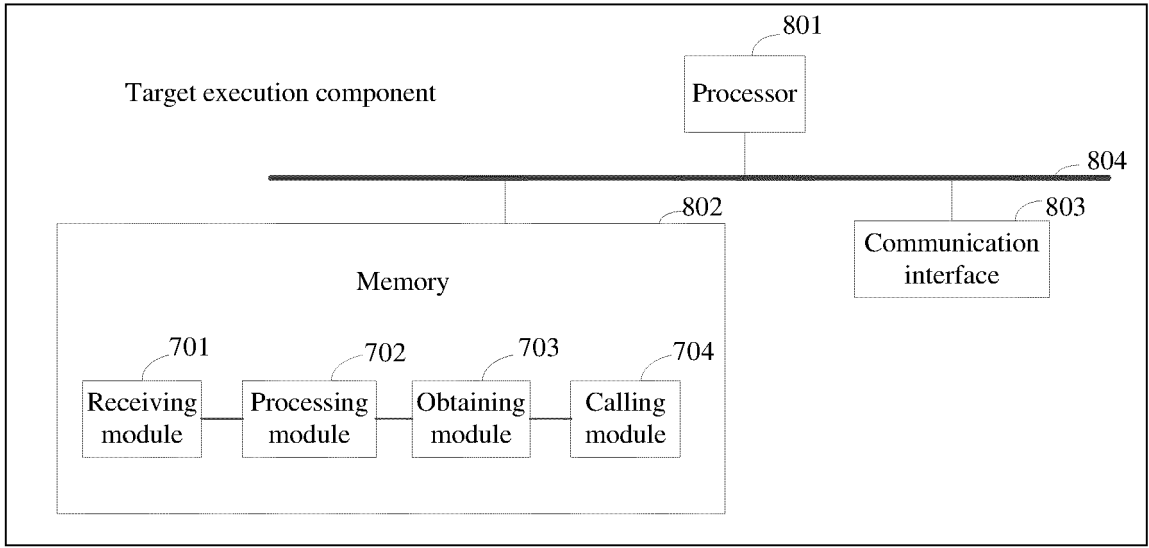


FIG. 8

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- US 20180359448 A1 [0005]